

Video viewing system and method

The invention relates to a video viewing system which supports time-shift viewing of a video stream.

5 PCT patent application WO99/33265 discloses a video viewing system which is capable of operating in a number of modes of operation, including a time shift mode, a live play mode and a pause mode. In the time-shift mode, an incoming video stream is recorded in a storage unit and in parallel therewith a significantly older (e.g. ten minutes older) part of the incoming stream is replayed to the output of the system from the storage unit. In the live play
10 mode, the incoming stream is fed directly to the output without any significant delay. In the pause mode, the system outputs a still image, which is the last image output during preceding live play and simultaneously records the incoming video stream in the storage unit.

Video mode switching between the different modes is not described in WO99/33265. In principle, video mode switching could be achieved by restarting the video
15 system each time when a video mode switch command is received, starting up data flows as required for the relevant mode. This, however, has the disadvantage that an interruption may occur in the video output signal, as a result of the need to fill the various pipe-lines in the system and to restart the decoding process, often starting from an arbitrary point of time in the incoming video stream which does not permit decoding to start immediately.

20 An alternative is to realize video mode switches by means of multiplexing. In this case, the video system keeps all tasks that need to be executed in any one of the modes (i.e. at least an input task, a recording task, a replay task and a decoding task) running in all modes. One or more multiplexers switch the sources of one or more of the tasks during a video mode switch. Because none of the tasks needs to be restarted, this potentially
25 eliminates interruptions. However, it has the disadvantage that certain tasks keep running also when they are not needed. In a video system, in which the tasks are implemented as software tasks running on a general-purpose processor, the processor resources occupied by the tasks remain unavailable to other processes. In a hardware-implemented system, the hardware that performs the tasks continues to consume unnecessary power supply current.

European patent application No. 01203905.3 (unpublished at the priority date of the present application and assigned to the same assignee) describes a mechanism for dynamically reconfiguring networks of processes or tasks.

5

It is, inter alia, an object of the invention to provide a video display system which is capable of operating in several modes of operation including a time-shift mode, wherein video mode switching can be realized without significant interruption and a task not needed during a certain mode need not continue execution in that certain mode.

10

The video display system according to the invention is set forth in claim 1. FIFO (First In First Out) communication buffers with reassignable sources and destinations are used for communicating video streams between the input, decoding, recording and replay tasks. During a mode switch, dynamic reconfiguration is realized by reassigning the source or destination of a FIFO communication buffer. The other side of the FIFO communication
15 buffer stays assigned to the same task before and after the mode switch. At the reassignment, video data may still be present in the FIFO communication buffer. The task that stays assigned to the FIFO communication buffer keeps on executing during the mode switch, as far as video data is available in the FIFO communication buffer. Thus, the use of FIFO communication buffers which remain active during the mode switch ensures that mode
20 switching can be performed without interruption of the video streams and without requiring tasks to remain active if they are no longer needed after the mode switch. The resources used by such a task may be freed after the mode switch.

In an embodiment, dynamic reassignment involves detachment and subsequent attachment of respective sources or destinations of video streams to the communication
25 buffer. Detachment and attachment occur in response to receiving a mode-switch command, but with delays until a subsequent boundary between closed Groups of Pictures has been detected in the detached and attached stream, respectively. A closed Group of Pictures in a video stream encodes a series of frames of video information that can be decoded without requiring access to frames outside the group. By delaying detachment and attachment, it is
30 prevented that "orphaned" video data occurs in the communication buffer, which cannot be decoded and might give rise to interruptions in the decoding task.

These and other objects and advantageous aspects will be described with reference to the following Figures.

Figure 1 shows a video display system,

Figures 2a-c show task graphs during different video modes.

5

Figure 1 shows a video display system. The system contains a video stream input 10, a video stream output 12, a plurality of processing task elements 14a-f, a plurality of FIFO communication buffers 16a-d, a buffer connection element 17, a storage device 18 and a switching control unit 15. A video display device 19 is coupled to output 12. Processing task elements 14a-f include an input task element 14a, a recording task element 14b, a replay task element 14c, a decoder task element 14f plus optional further elements 14d,e. Input 10 is coupled to input task element 14a. Output 12 is coupled to decoder task element 14f. Storage device 18 is coupled to recording task element 14b and replay task element 14c. Processing task elements 14a-f are coupled to FIFO communication buffers 16a-d via buffer connection element 17. Switching control unit 15 is coupled to processing task elements 14a-f, FIFO communication buffers 16a-d and buffer connection element 17.

In a first embodiment, each component shown in Figure 1 is a dedicated hardware element. In other embodiments, one or more of the components may be implemented in a general purpose processor by means of suitable computer programs. For example, in the first embodiment, processing task elements 14a-f are each different hardware elements with a design dedicated to the task to be performed by the processing task element 14a-f, but in the other embodiments part or all of the processing task elements 14a-f may be implemented as different computer programs loaded in a general purpose processor. Similarly, in the first embodiment, FIFO communication buffers 16a-d are hardware FIFO buffers with data inputs, data outputs, and buffer full and empty signaling outputs. However, in the further embodiments, FIFO communication buffers 16a-d may be implemented as different buffer areas in a processor memory with appropriate software for signaling to task elements 14a-f whether or not the buffers are full or whether or not they are empty. Also similarly, in the first embodiment, buffer connection element 17 may be a hardware element such as a switch matrix and in the further embodiments buffer connection element 17 may be implemented as an interface program between FIFO communication buffers 16a-d and processing task elements 14a-f. Furthermore, switching control unit 15 may be implemented in hardware or as a software programmed computer.

In operation, an incoming video stream is applied to input 10 and an outgoing video stream is produced at output 12, for display at video display device 19. In between, processing task elements perform various processing tasks on video data that derives from the incoming stream.

5 Video mode control unit 15 controls the video mode in which the video viewing system operates. The combination of tasks executed depends on the video mode in which the system operates. The video modes include a live play mode, a pause mode and a time-shift mode.

Figures 2a-c show task graphs of the processing in different video modes.

10 Figure 2a shows operation in the live play mode, in an embodiment where the incoming stream is an analog video signal. In this case, an input task 20 is performed which receives the incoming stream and outputs an encoded stream to a decoding task 22 via a first connection 21. Decoding task 22 outputs a decoded stream for use by video display device 19 (not shown). Although a single decoding task 22 has been shown, it will be appreciated that
15 the decoding task may in fact be comprised of several tasks, such as a demultiplexing task for demultiplexing a program from an MPEG transport stream, followed by a program decoding task operating on the demultiplexed program. Similarly separate video and audio stream processing tasks may occur. For reasons of simplicity, only a single task is shown.

Figure 2b shows operation in the pause mode. In this mode, a recorder task 24
20 has been added. Input task 20 continues to be performed and outputs the encoded stream to recorder task 24 via first connection 21. Recorder task 24 records the encoded stream in storage device 18 (not shown). Decoder task 22 has been switched to a "freeze" state, in which it permanently outputs its last received video frame, without reading from its input. Its input connection has been uncoupled. In an alternative embodiment, in which decoder task
25 22 is arranged to permanently output its last full received frame when it finds that no new frame has been supplied to its input, first connection 21 may be left connected to decoder task 22, a second connection being added between input task 20 and recording task 24. In this alternative embodiment, first connection 21 may be disconnected from input task 20.

Figure 2c shows operation in the time-shift mode. In this mode, the input task
30 20, the decoder task 22 and the recorder task 24 continue to be performed and a replay task 26 has been added. Input task 20 outputs the encoded stream to recorder task 24 via first connection 21. Replay task 26 outputs a delayed version of the input stream from storage device 18 (not shown) to decoder task 22 via a second connection 23. Decoder task 22 has

been switched back to the normal operating state, in which it reads video data from its input and uses that data to update its output.

In the apparatus of Figure 1, tasks 20, 22, 24, 26 are executed by processing task elements 14a-c,f. Connections 21, 23 are implemented by FIFO communication buffers 16a-d. Tasks that are not required in a video mode are deactivated, for example, by disabling the clock input to the processing task element in the case of hardware-implemented task elements, or by releasing resources used by these tasks in the case of software-implemented task elements, thus releasing system resources for other uses.

When video mode control unit 15 switches the video display system between different video modes, the processing task elements 14a-c,f that are active before and after the mode switch are kept active (or at most suspended) through the mode switch, so that they will continue execution after the mode switch. The FIFO communication buffers 16a-d coupled to these continuously active processing task elements 14a-c,f also remain active, but video mode control unit 15 controls buffer connection element 17 so that the coupling to or from some of the inputs or outputs of FIFO communication buffers 16a-d are switched to different processing task elements 14a-c,f.

When a first one of processing task elements 14a-c,f has an input or output coupled, via a FIFO communication buffer 16a-d, to a second one of processing task elements 14a-c,f before the mode switch and to a third one of processing task elements 14a-c,f after the mode switch, the second one of the processing task elements 14a-c,f is decoupled from this FIFO communication buffer 16a-d and the third one is coupled to this FIFO communication buffer 16a-d in its place during the video mode switch. Video data in the FIFO communication buffer 16a-d is left in place for later use. The second one of processing task elements 14a-c,f is deactivated (or its execution suspended) before being decoupled and the third one is activated or unsuspended after being coupled (or, if activated earlier, it is supplied with a dummy "FIFO full" signal if its output is coupled to the FIFO communication buffer 16a-d or "FIFO empty" signal if its input is to be coupled). Thus, no data is lost during mode switching and there is no interruption of the video signals.

In the case of a switch from the live play mode of Figure 2a to the pause mode of Figure 2b, video mode switch control unit 15 sends a signal to the decoding task element 14f to switch to the "freeze" state of operation, causing it to repeat a current frame. Video mode switch control then disconnects the FIFO communication buffer 16a-d that implements connection 21 from decoding task element 14f and connects it to a recording task element

14b. Subsequently, video mode switch control unit 15 activates the recording task element 14b.

In the alternative embodiment, in which first connection 21 remains coupled to decoder task 22, video mode switch control unit 15 reconnects the output of input task
5 element 14a to a second FIFO communication buffer 16a-d (optionally causing input task 20 to suspend operation during reconnection) and connects recording task element 14b to that second FIFO communication buffer 16a-d before activating recording task element 14b. In this embodiment, decoder task 22 does not have to be brought into a freeze state. This simplifies design, but it has the disadvantage that some delay may occur during video mode
10 switching because buffered data from first connection 21 has to be processed before decoder task 22 outputs a frozen image.

In the case of a switch from the pause mode of Figure 2b to the time-shift mode of Figure 2c, video mode switch control unit 15 connects a replay task element 14c and the decoding task element 14f to a second FIFO communication buffer 16a-d that implements
15 connection 23. Subsequently, video mode switch control unit 15 signals decoding task element 14f to return to the normal state of operation, in which it reads video data for new frames from second FIFO communication buffer 16a-d, and activates the replay task element 14b. In the alternative embodiment, wherein the first connection 21 remains attached to decoding task 22, no new connection needs to be added of course. In this case, video mode
20 switch control unit 15 couples replay task element 14c to the FIFO communication buffer 16a-d that implements first connection 21 and activates replay task element 14c. No signal is needed to switch decoder task element 14f back to the normal operating state in this case.

In the case of a switch back from the time shift mode of Figure 2c to the live play mode of Figure 2a, video mode switch control unit 15 signals replay task element 14c to
25 cease execution. Video mode switch control unit 15 commands recording task element 14b and replay task element 14c to cease execution. Then video mode switch control unit 15 reconnects the input of decoding task element 14f from the FIFO communication buffer 16a-d that is connected to replay task element 14c to the FIFO communication buffer 16a-d that implements first connection 21 to input task element 14a. Preferably, a signal is sent to
30 switch decoder task element 14f to the "freeze" mode temporarily during reconnection. Afterwards, the FIFO communication buffer 16a-d implementing connection 23 may also be released and its contents discarded.

Alternatively, input task element 14a may be coupled to the input of the FIFO communication buffer 16a-d that couples replay task element 14c to decoder task element 14f

in the time shift mode. However, this has the disadvantage that switching from time shift replay to live play occurs only with some delay, because decoder task element 14f has to read the old video data from the FIFO communication buffer 16a-d first.

In the alternative embodiment, wherein decoder task 22 remains coupled to first connection 21, video mode switch unit 15 reconnects the input of decoder task 22 to the FIFO communication buffer 16a-d that implements second connection 23. It will be seen that thus, through switching from the live play mode to the pause mode, to the time shift mode and back to the live play mode, a different one of FIFO communication buffers 16a-d is coupled to encoder task element 14f compared with the initial live play mode. This is a result of dynamic assignment of the FIFO communication buffers 16a-d.

It will be appreciated that in this way no restart of any processing task elements 14a-f is needed when the element is active both before and after the video mode switch. Processing task elements 14a-f that are no longer needed are deactivated, releasing system resources for other use. Continued use of FIFO communication buffers 16a-d before and after the video mode switch ensures continuity of the viewed video streams. It will also be appreciated that a similar implementation can be used for the reverse of the switches that have been described (i.e. switches from time shift mode to pause mode, from pause mode to live play mode and from live play mode to time shift mode).

Preferably, the processing task elements 14a-f are designed to select the point of time where they cease to be active dependent on the progress of the video data being processed, so that the part of the stream after ceasing can be decoded without reference to earlier video data. In an MPEG video signal, for example, I-frames, P-frames and B-frames occur, the latter two types of frame being encoded as updates to other frames. Closed "Groups Of Pictures" (GOPs) are identified in the video stream, so that the frames in each particular GOP are never encoded as updates to frames outside that particular GOP.

In this case, when video mode control unit 15 signals to a processing task element 14a-f to suspend or cease operation as part of a video mode switch, the processing task element 14a-f finishes reading and/or writing of a GOP from or to a FIFO communication buffer 16a-d prior to actually ceasing or suspending operation. Thereupon the processing task element 14a-f signals back to video mode switch control unit 15 that the command has been executed. Subsequently, video mode switch control unit 15 signals buffer connection element 17 to couple the buffer or buffers involved to another processing task element 14a-f as required by the mode switch. After that, video mode control unit signals the newly connected processing task element 14a-f to start or resume execution.

FIFO communication buffers 16a-d preferably provide video data input, video data output and empty/full signaling. The data width is not relevant to the invention but typically multibit datawords will be written and read. Each FIFO communication buffer 16a-d provides a full/not full signal back to the processing task element 14a-f that supplies data to the buffer. If "full" is signaled and the processing task element 14a-f has data available, the processing task element 14a-f suspends execution. Each FIFO communication buffer 16a-d provides an empty/not empty signal to the processing task element 14a-f that reads data from the buffer. If "empty" is signaled and the processing task element 14a-f needs to read data, the processing task element 14a-f suspends execution. In the software implementation of processing task element 14a-f, the processing task element 14a-f may use polling of the empty and full signals to determine when to resume operation, or event signaling or interrupt signaling may be used to resume operation. In the hardware implementation, the empty and full signals may also be used as enable or disable signals.

FIFO communication buffers 16a-d may be implemented as memory areas in a computer memory, the FIFO operation being controlled by software (keeping a pointer to the oldest unread data and the free location next to last written data, checking for full and empty conditions and setting flags accordingly or signaling those flags to coupled processing task elements 14a-f). Dedicated hardware FIFO buffers may be used as well. Similarly, buffer connection element 17 may be implemented using software, in which case buffer connection element 17 may be integrated with FIFO communication buffers 16a-d in the form of pointers to processing task elements 14a-f to which respective FIFO communication buffers 16a-d are coupled. Alternatively, buffer connection element 17 may be integrated with processing task elements 14a-f in the form of pointers to FIFO communication buffers 16a-d to which respective processing task elements 14a-f are coupled. Mixtures of such pointer implementations are also possible.

It will be appreciated that the invention is not limited to the specific embodiments that have been described.